

La funcion del ejercicio 4 crea una composicion de 2 imagenes A y B, donde A es una imagen transformada segun la matriz de transformacion T.

```
function [] = ejercicio_4(A,B,T)
```

Una vez recibido por parametro las 2 imagenes y la matriz de transformacion, la primera tarea a realizar es transformar la imagen A:

```
A_trans = imwarp(A,T);
```

Una vez transformada, debemos componer la imagen. Para crear la composicion, ambas imagenes deben tener el mismo tamaño. Hemos decidido redimensionar la imagen A ya transformada y para ello, necesitamos tanto sus dimensiones como las de la imagen B:

```
[rowB, colB, ~] = size(B);  
[rowA, colA, ~] = size(A_trans);
```

Para evitar deformaciones, y aprovechar al maximo el espacio, forzaremos unicamente una de las dimensiones de la imagen y la otra la extenderemos con un fondo negro. Forzaremos aquella dimension que sea mayor a la otra. Para ello comparamos las relaciones de aspecto. Si la relacion de aspecto de la imagen A transformada es menor que la de B, significa que A es mas alta (en proporcion) que B y por lo tanto, ajustaremos la altura u extenderemos con negro horizontalmente. En caso contrario, ajustaremos la anchura, y extenderemos con negro verticalmente:

```
if (colA/rowA < colB/rowB) %mas vertical A que B  
    disp("Modo 1");
```

Con este remidimensionamiento, el programa mantiene la relacion de aspecto, forzando en este caso la altura, y adaptando la anchura.

```
AR = imresize(A_trans, [rowB NaN]);  
[~, colAR, ~] = size(AR);
```

En este momento, deberemos extender en negro lo necesario en la otra dimension. A partir de este momento, la imagen A transformada tiene exactamente las mismas dimensiones que B y ya se podria crear la composicion.

```
AR(1,end+(colB-colAR),1) = 0;
```

Pero nosotros hemos creido conveniente desplazar la imagen al centro para que no se quedara pegada al borde de la imagen.

```
AR = imtranslate(AR, [((colB-colAR)/2) 0 0]);
```

El siguiente codigo, es analogo al anterior. Fuerza la anchura y extiende verticalmente.

```

else
    disp("Modo 2");
    AR = imresize(A_trans, [NaN colB]);
    [rowAR, ~, ~] = size(AR);
    AR(end+(rowB-rowAR),1,1) = 0;
    AR = imtranslate(AR, [0 ((rowB-rowAR)/2) 0]);
end

```

Finalmente, creamos la composicion. En este caso nosotros hemos decidido dejar la imagen B tal y como esta, y colocarle encima la imagen A transformada con una opacidad del 50%, es decir, creando un efecto de transparencia.

```

C = B + 0.5*AR;
imshow(C);
end

```

Output del ejercicio 4:

```
A = imread('..\Media\imatgeA.jpg');  
B = imread('..\Media\imatgeB.jpg');
```

En este caso realizaremos 2 ejecuciones distintas con las mismas imagenes cambiando la matriz de transformacion.

- En la primera ejecucion mostramos el output tras una transformamos neutra a la imagen A:

```
T = affine2d([1 0 0; 0 1 0; 0 0 1]);  
ejercicio_4(A,B,T);
```

Ejercicio 4:

Concatena la imagen A transformada segun la matriz de transformacion T con la imagen B.

Modo 1



- En la segunda ejecucion mostramos el output con la imagen A transformada por una matriz no neutra.

```
T = affine2d([1 0 0; 0.5 1 0; 0 0 1]);  
ejercicio_4(A,B,T);
```

Ejercicio 4:

Concatena la imagen A transformada segun la matriz de transformacion T con la imagen B.

Modo 1

